A HAPTIC ENABLED UML CASE TOOL

Atif Alamri, Mohamad Eid, and Abdulmotaleb El Saddik
Multimedia Communications Research Laboratory
University of Ottawa, Canada
E-mail:{atif, eid, abed}@mcrlab.uottawa.ca

ABSTRACT

This paper describes a haptic enabled UML CASE tool that enables software engineering developers to physically manipulate and touch the UML modeling elements and feel the force feedback. We propose an architecture and a software design for the tool. The current implementation of the tool uses the Omni Phantom device, a quite common haptic interface among the haptic research community. The CASE tool, from a user perspective, consists of three parts: a drawing area, a palette, and a tool bar. Our preliminary usability study demonstrated the potential of adding the haptic modality to UML development tools.

1. INTRODUCTION

Haptics, a term which was derived from the Greek verb "haptesthai" meaning "to touch", refers to the science of touch and force feedback in human-computer interaction. Currently, research on haptics is categorized into human haptics, machine haptics, computer haptics, and multimedia haptics [1]. In recent years the haptics technology has been gaining increasing interest in both academia and industry. Its applications are spread in games, arts and media creation, manufacturing, Tele-robotics and Tele-operation, education and training, as well as medical simulation and rehabilitation. Examples of haptic-based applications can be found in [2-4]. On top of the visual modality, haptics modality can be considered as an extra channel of communication that helps software developers design UML models in intuitive and entertaining manner.

A UML CASE tool is a software system that supports the notations and semantics, e.g., the creation and editing of diagrams, associated with the Unified Modeling Language (UML) [5]. Additionally, it supports code generation from models and model generation from code (reverse engineering). Current UML CASE tools depend on standard input/output devices such as mouse, keyboard, and recently whiteboards, to facilitate users' interactions. Even though the large size and ease of use of whiteboards improve the development efficiency, these tools are not social. This can greatly influence these tools' ability of intuitivism and collaboration. Haptics, on the other hand, can greatly improve the realism of the interaction between developers and the modeling elements in a UML CASE tool.

Haptics offer a natural user interface based on the human gesture system. The resistance and friction provided by force feedback devices add a natural and intuitive feel of everyday tasks such as dragging, sliding levers, and pressing/depressing buttons. Furthermore, a haptic-enhanced tool may generate special force feedback stimuli that guide/warn the user when trying to perform inappropriate modeling activities. Haptic devices may also help in standardizing the way developers interact with UML case tools by using agreed upon physical stimuli. The user will be able to sense the texture and stiffness of the modeling elements he/she models. Finally, the possibility of palpating classes, objects and components is expected to add some sort of excitement to the modeling process.

In this paper, we present the system architecture and a software design models of the proposed haptic-based UML case tool. The rest of the paper is organized as follows: section 2 explores related work in the field and highlights the novelty of our proposed tool. Section 3 describes the tool compartments and elements, and defines the functionalities and scope of the tool. In addition, it explains the haptic stimuli that were implemented in the system. Section 4 discusses the tool software architectural design. Section 5 explores the implementation details and the technologies and tools employed to develop the system's software. In section 6, we explain the usability study we performed using the tool and discuss our findings. Finally, we conclude and summarize the paper in section 7 and propose possible future work.

2. RELATED WORK

The incorporation of Haptics in standard graphical user interfaces (GUIs) has shown significant potential as a substitute sensory for visually impaired or blind users. For instance, the Moose prototype system [6] enables blind users to navigate and interact with a desktop application through a GUI window. By moving a haptic mouse over a GUI control object (such as a button, check box, menu, etc.) the user feels differently and eventually can identify the type and current state of the controls (such as the state of a check

1-4244-1017-7/07/$25.00 ©2007 IEEE

box). Another application examined the use of three designs of a scrolling function in virtual environment [7]. The user presses the slide of a world-limiting box and the world behind it moves into the limiting box. The user may also use the keyboard to move the world in a specific direction, or use the Phantom stylus button to drag the world along the horizontal plane. In [8], a user-friendly haptic environment has been developed to allow blind or visually impaired people to access interactive presentations based on HTML Web pages. The application has both tactile and audio feedback.

Ian Oakle and colleagues [9] developed CHASE, a collaborative haptic-based tool for structured editing and drawing. The tool has been used as a simple CASE tool. Participants were asked to read a problem statement and design a set of UML diagrams to solve it. CHASE allows users to create and edit four types of object: text items, rectangular groups, oval groups, and links. However, this work is different from our work in the sense that the designed tool was a free diagramming tool that facilitate the concepts of shared editors rather than a CASE tool. We stress that the novelty of the presented work is in exploring and studying a new field of haptic technology application by incorporating haptic properties into UML modeling and developing a light-weight CASE tool with an integrated mechanism to translate these properties into force feedback and stiffness the user can observe and feel.

3. THE TOOL DESCRIPTION

UML CASE tools are complex software systems which incorporate exhaustive features and functionalities. However, we intended to limit our implementation to the portion that will prove the concept and confirm the feasibility of the application. Therefore, we have decided, for the time being, to limit the implementation of the tool to a haptic-enabled class diagram. The CASE tool consists of three parts: a drawing area, a palette, and a tool bar. The palette contains icons for the basic modeling elements that the user can manipulate through the haptic device. Basically, two main types of modeling elements are required to build a class diagrams: the class and the relation. The tool bar provides the user with basic filling functionalities he/she is allowed to apply to the class diagram. The drawing area contains the user drawn class diagram. The implementation of the drawing area is not easy because it includes the haptic interactions as well.

Abstractly, the functionalities of the tool are decomposed into two groups: class diagram filing and class diagram manipulation. The former includes basic filing functionality such as new, open, save, and print whereas the latter contains basic requirements for building class diagrams and here is where haptics integration resides. Figure 1 shows a use case diagram for class diagram manipulation along with the types of expected interactions and events. The developer can create classes and relations, delete them, and move them in the drawing area.

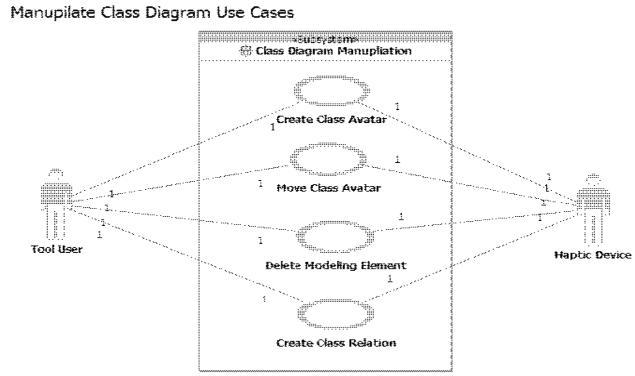

Figure 1: A use case diagram for diagram manipulation

Mainly, there are three types of haptic stimuli that were implemented in the tool:

- **Class Weight:** This stimulus enables users to feel the weight of the class depending on the number of its members (class attributes and methods). This allows developers to intuitively manipulate the UML class shapes and feel their contents.
- **Relationship Elasticity:** This stimulus is fired when the user creates a relationship between two classes in the drawing area. The user feels an elastic perception while he/she is creating a UML relation to make the modeling process more realistic and exciting.
- **Class Collisions:** Whenever the tool detects an overlap between existing classes, it sends back a repulsive force feedback that resists the collision action. This makes the user design more sorted and be enforced to modulate his/her design.

4. THE TOOL ARCHITECTURE DESIGN

In this section, we describe the software architecture of the tool. Figure 2 shows a component view of the class diagram manipulation, where the haptic rendering is incorporated. Haptic rendering component contains design compartments that implement the haptic rendering process. It comprises three parts: the collision detection, the force feedback computation, and the control algorithm. Graphic rendering component contains design elements for detecting collision and computing the penetration depth for colliding objects. The haptic device component interfaces the haptic device API. The current implementation of the tool uses the PHANToM Omni device [10]. The interfaces of both devices reside in this component.

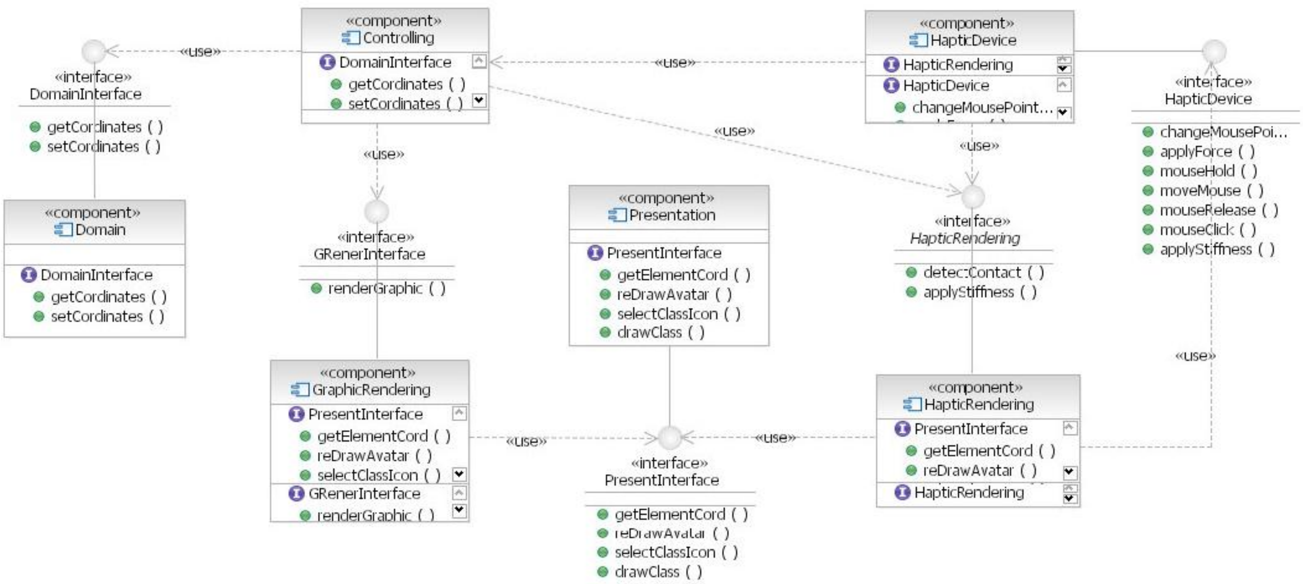

Figure 2: A component view of the class diagram manipulation

To capture the tool functionalities and enforce Model-View-Controller (MVC) design pattern we added three more components. As shown in Figure 2, the presentation component contains the tool's Graphical User Interface (GUI) and is used by the haptic rendering and the graphic rendering components. The controlling component includes classes that are responsible for orchestrating the events mentioned earlier in the use case diagram (see Figure 1). The domain component captures the UML modeling elements created by the tool and ensures UML class diagram integrity by mastering a set of diagram rules.

Whenever the user grabs a class and starts moving it, the tool gives the user the perception of holding the class (by associating a gravitational weight that is proportional to the contents of the class). This is done through the haptic rendering component. The tool updates its interface with the new position of the class through the presentation component. Once the user releases the object, the tool halts the gravitational perception and updates its GUI with the final position of the object. In case the user moves a class over another one, the tool controls the proper force response through the graphic rendering component.

5. IMPLEMENTATION

The implementation has employed different tools and utilized multiple technologies (see Figure 3). The Rational Software Architect (RSA) has been used for modeling the software architecture of the tool [11]. Eclipse is the platform that was used for coding and building the tool [12]. We used Eclipse because of its extendibility features. Eclipse uses Standard Widget Toolkit (SWT) and JFaces to display GUI elements. SWT is written in Java and presents an alternative to the standard (AWT and Swing) graphical libraries of Java. JFace is a user interface (UI) toolkit with classes for handling many common UI programming tasks. The toolkit simplifies the construction of applications based on SWT and provides actions to define the user's behavior and eventually assigns that behavior to specific components.

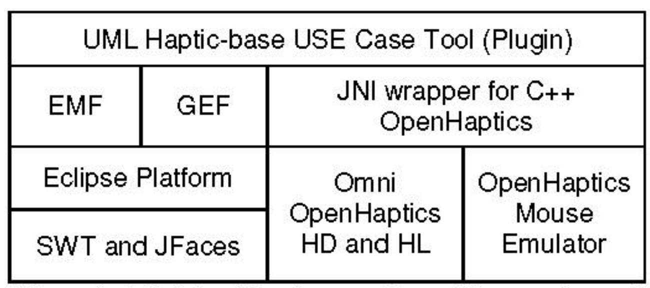

Figure 3: A high level implementation architecture for tool

The GUI parts of the tool such as drawing area, palette and project explorer are provided by the eclipse platform. The Eclipse Modeling Framework (EMF) is used to provide the UML modeling and diagramming facility whereas the Graphical Editing Framework (GEF) is used to facilitate the display of any graphic model and support interactions from mouse and/or keyboard. All graphical visualization is done via the Draw2D framework, which is a standard 2D drawing framework based on SWT from eclipse.

The haptic device is made accessible for programmers through the OpenHaptics API (HDAPI and HLAPI) [13]. The HDAPI is a low-level foundation layer that is suited for developers familiar with haptic paradigms and rendering.

The HLAPI is built on top of the HDAPI and supports high level scene rendering. It is desired for quick and easy haptic applications development. We used the JNI wrapper to wrap the OpenHaptics API and invoke native API methods.

6. EVALUATION OF THE TOOL

The objective of the evaluation is to determine the quality of the designed haptic-enabled UML tool compared to mouse-based tools. We have designed a simple class diagram and asked participants to perform the modeling task using the PHANToM Omni device in two different settings. First, we disabled the haptic force feedback from the device, while in the second set of experiments; we enabled the force feedback capabilities of the haptic device. The test subjects were asked to practice and utilize the three haptic stimuli defined above (class weight, relationship elasticity and class collision) during their interaction with the tool. Users' experience are measured and compared by exploiting a questionnaire that reflects the personal opinion on the tool. Ten participants took part in the usability study, most of which are familiar with UML modeling.

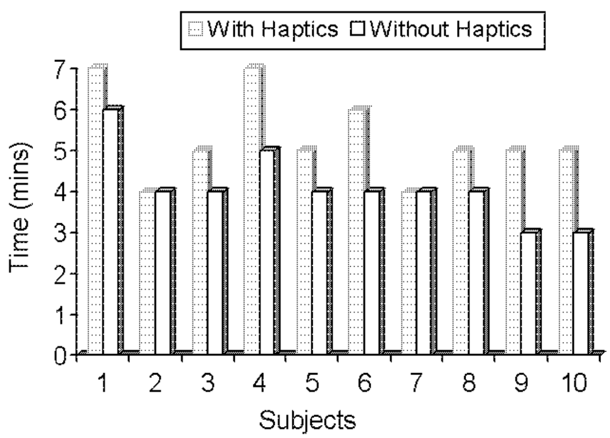

Figure 4. Task completion task for 10 subjects

We found that the task completion time was larger when haptic feedback was used (an average of 5.2 mins. with haptic feedback and 4.1 mins. using traditional mouse, see Figure 4 for details). The reason is that users are not used to haptic interface compared to regular mouse. However, 90% of the subjects agreed that the haptic stimuli have increased the interactivity and entertainment while developing a UML diagram. As per the simulated stimuli, 50% of the subjects found that gravity was the most perceivable, 30% considered elasticity the most useful one, and 20% perceived the collision effect as the most important one. Also, holding the device arm has been found tiresome due to its inertia.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the development of a haptic-enabled UML CASE tool. Then, we explored the system's architecture and the software design for this tool. Three haptic stimuli were simulated: class gravity, relationship elasticity, and collision detection. The usability study showed that, in many cases, haptics enhances the interaction between a UML developer and the CASE tool. As per future work, our intention is to include other types of UML modeling diagrams.

8. ACKNOWLEDGMENT

This research is partially supported by NSERC and ORNEC. Also we would like to acknowledge the support from IBM Ottawa Software Lab.

9. REFERENCES

[1] M. Eid, M. Orozco and A. El Saddik, "A Guided Tour in Haptic Audio Visual Environment and Applications", International Journal of Advanced Media and Communication, vol.1, n.3, 2007.

[2] A. El Saddik, M. Orozco, Y. Asfaw, S. Shirmohammadi, and A. Adler, "A Novel Biometric System for Identification and Verification of Haptic Users", IEEE Transactions on Instrumentation and Measurement, to appear.

[3] B. Malek, M. Orozco, and A. El Saddik, "Novel Shoulder-Surfing Resistant Haptic-based Graphical Password", EuroHaptics 2006 conference, July 3-6 Paris, France.

[4] I. Shakra, M. Orozco, A. El Saddik, S. Shirmohammadi, and E. Lemaire, "Haptic Instrumentation for Physical Rehabilitation of Stroke Patients", MeMeA, 2006.

[5] Unified Modeling Language Resource Page: www.uml.org.

[6] S. O'Modhrain and R. B. Gillespie. "The Moose: A Haptic User Interface for Blind Persons", In Proceedings of the Third WWW6 Conference, Santa Clara, CA, April 1997.

[7] C. Magnuson and K. Rassmus-Gröhn, "Non-visual Zoom and Scrolling Operations in a Virtual Haptic Environment", Eurohaptics 2003, Dublin, Ireland, 2003.

[8] G. Nikolakis, I. Tsampoulatidis, D. Tzovaras, and M.G. Strintzis, "Haptic browser: a haptic environment to access HTML pages", In SPECOM-2004, 514-520, 2004.

[9] I. Oakley, S. Brewster P.D. Gray, "Can You Feel the Force? An Investigation of Haptic Collaboration in Shared Editors" EuroHaptics'2001, Birmingham, UK, 2001.

[10] PHANTOM® Omni™ Haptic Device: http://www .sensable.com/products/phantom_ghost/phantom-omni.asp.

[11] Rational Software Architect RSA. http://www-306 .ibm.com/software/awdtools/architect/swarchitect/index.html

[12] http://www.eclipse.org

[13] http://www.sensable.com/products-openhaptics-toolkit.htm.